%% 第9章 蚁群算法及MATLAB实现——TSP问题

% 程序9-1

%% 数据准备

% 清空环境变量

clear all

clc

% 程序运行计时开始

t0 = clock;

% 导入数据

citys = xlsread('D:\berlin52.xlsx','B2:C53');

%% 计算城市间距离

n = size(citys,1);

D = zeros(n,n);

for i = 1:n

for j = 1:n

if i ~= j

D(i,j) = sqrt(sum( ( citys(i,:) - citys(j,:) ).^2 ) );

else

D(i,j) = 1e-4; %设定的对角矩阵修正值

end

end

end

%% 初始化参数

m = 31; % 蚂蚁数量

alpha = 1; % 信息素重要程度因子

beta = 5; % 启发函数重要程度因子

vol = 0.2; % 信息素挥发(volatilization)因子

Q = 10; % 常系数

Heu\_F = 1./D; % 启发函数(heuristic function)

Tau = ones(n,n); % 信息素矩阵

Table = zeros(n,n); % 路径记录表

iter = 1; % 迭代次数初值

iter\_max = 100; % 最大迭代次数

Route\_best = zeros(iter\_max,n); % 各代最佳路径

Length\_best = zeros(iter\_max,1); % 各代最佳路径的长度

Length\_ave = zeros(iter\_max,1); % 各代路径的平均长度

Limit\_iter = 0; % 程序收敛时迭代次数

%% 迭代寻找最佳路径

while iter <= iter\_max

% 随机产生各个蚂蚁的起点城市

start = zeros(m,1);

for i = 1:m

temp = randperm(n);

start = temp(1);

end

Table(:,1) = start;

% 构建解空间

citys\_index = 1:n;

% 逐个蚂蚁路径选择

for i =1:m

% 逐个城市路径选择

for j = 2:n

tabu = Table(i,1:(j - 1)); % 已访问的城市集合（禁忌表）

allow\_index = ~ismember(citys\_index,tabu); % 参数说明1（程序底部）

allow = citys\_index(allow\_index); % 待访问的城市集合

P = allow;

% 计算城市间转移概率

for k = 1:length(allow)

P(k) = Tau(tabu(end),allow(k))^alpha \* Heu\_F(tabu(end),allow(k))^beta;

end

P = P / sum(P);

% 轮盘赌法选择下一个访问城市

Pc = cumsum(P); % 参数说明2（程序底部）

target\_index = find(Pc >= rand);

target = allow(target\_index(1));

Table(i,j) = target;

end

end

% 计算各个蚂蚁的路径距离

Length = zeros(m,1);

for i = 1:m

Route = Table(i,:);

for j = 1:(n - 1)

Length(i) = Length(i) + D(Route(j),Route(j + 1));

end

Length(i) = Length(i) + D(Route(n),Route(1));

end

% 计算最短路径距离及平均距离

if iter == 1

[min\_Length, min\_index] = min(Length);

Length\_best(iter) = min\_Length;

Length\_ave(iter) = mean(Length);

Route\_best(iter,:) = Table(min\_index,:);

Limit\_iter = 1;

else

[min\_Length,min\_index] = min(Length);

Length\_best(iter) = min(Length\_best(iter - 1),min\_Length);

Length\_ave(iter) = mean(Length);

if Length\_best(iter) == min\_Length

Route\_best(iter,:) = Table(min\_index,:);

Limit\_iter = iter;

else

Route\_best(iter,:) = Route\_best((iter - 1),:);

end

end

% 更新信息素

Delta\_Tau = zeros(n,n);

% 逐个蚂蚁计算

for i = 1:m

% 逐个城市计算

for j = 1:(n - 1)

Delta\_Tau(Table(i,j),Table(i,j+1)) = Delta\_Tau(Table(i,j),Table(i,j+1)) + Q/Length(i);

end

Delta\_Tau(Table(i,n),Table(i,1)) = Delta\_Tau(Table(i,n),Table(i,1)) + Q/Length(i);

end

Tau = (1 - vol) \* Tau + Delta\_Tau;

% 迭代次数加1,清空路径记录表

iter = iter + 1;

Table = zeros(m,n);

end

%% 结果显示

[Shortest\_Length,index] = min(Length\_best);

Shortest\_Route = Route\_best(index,:);

Time\_Cost = etime(clock,t0);

disp(['最短距离:' num2str(Shortest\_Length)]);

disp(['最短路径:' num2str( [Shortest\_Route Shortest\_Route(1)] )]);

disp(['收敛迭代次数:' num2str(Limit\_iter)]);

disp(['程序执行时间:' num2str(Time\_Cost),'秒']);

%% 绘图

figure(1)

plot([ citys(Shortest\_Route,1);citys(Shortest\_Route(1),1) ], [ citys(Shortest\_Route,2);citys(Shortest\_Route(1),2) ], 'o-');

grid on;

for i = 1:size(citys,1)

text(citys(i,1),citys(i,2),[' ' num2str(i)]);

end

text(citys(Shortest\_Route(1),1),citys(Shortest\_Route(1),2),' 起点');

text(citys(Shortest\_Route(end),1),citys(Shortest\_Route(end),2),' 终点')

xlabel('城市位置横坐标');

ylabel('城市位置纵坐标');

title(['ACA最优化路径(最短距离:' num2str(Shortest\_Length) ')']);

figure(2);

plot(1:iter\_max,Length\_best,'b');

legend('最短距离');

xlabel('迭代次数');

ylabel('距离');

title('算法收敛轨迹');

%% 程序解释或说明

% 1.ismember函数判断一个变量中的元素是否在另一个变量中出现,返回0-1矩阵；

% 2.cumsum函数用于求变量中累加元素的和,如A=[1,2,3,4,5],那么cumsum(A)=[1,3,6,10,15]。